Application of machine learning to pyrolysis reaction networks: reducing model solution time to enable process optimization

Blake R. Hough¹, David A.C. Beck^{1,2}, Daniel T. Schwartz¹, Jim Pfaendtner^{1,*}

¹ Department of Chemical Engineering, University of Washington, Seattle, WA, USA

² eScience Institute, University of Washington, Seattle, WA, USA

* Jim Pfaendtner, Department of Chemical Engineering, University of Washington, Box 351750, Seattle, WA 98195. E-mail: jpfaendt@uw.edu

Keywords: neural network, kinetic modeling, pyrolysis, biomass

Abstract

Comprehensive models of biomass pyrolysis are needed to develop renewable fuels and chemicals from biomass. Unfortunately, the detailed kinetic schemes required to optimize industrial biomass pyrolysis processes are too computationally expensive to include in models that account for both kinetics and transport within reacting particles. Here we present a machine learning approach using artificial neural networks and decision trees to reduce the computational expense of detailed kinetic models by four orders of magnitude, enabling their use in comprehensive models. The trained neural networks generalize very well, predicting the outputs of the detailed kinetic model with over 99.9% accuracy on new data. The machine learning approach we outline is not specific to kinetic modeling and can be applied to any set of input and output data, even if the underlying relationship between inputs and outputs is unknown.

1. Introduction

Interest in developing comprehensive models that describe biomass pyrolysis is high due to growing scientific and industrial interest in lignocellulosic biomass as a feedstock for renewable fuels and chemicals.(Ragauskas et al., 2006) These models couple descriptions of chemical kinetic mechanisms for biomass degradation with conservation equations for heat and mass transport.(Di Blasi, 2008; Moghtaderi, 2006) Numerical solutions to such complex models, especially in two- or three-dimensions, are computationally expensive. To reduce the computational cost, published comprehensive models incorporate many simplifying assumptions for both the kinetic and transport processes that tend to limit the predictive capabilities of the model. This is especially true for those models that use CFD approaches to move beyond single particles and model whole reactors.(Anca-Couce, 2016; Anca-Couce & Zobel, 2012; Mehrabian et al., 2012; Moghtaderi, 2006; Papadikis et al., 2009; Paulsen et al., 2014)

Dynamic mathematical models are commonly used in process engineering across disciplines as key tools for model-based control, product and process design, and optimization. In particular, detailed models based on the underlying physics of a scientific phenomenon can serve as tools that complement, and, in some cases, replace physical experiments. Model predictions can often provide additional information that is difficult or impossible to measure, and are typically less expensive and take less time than experiments. Even so, in many cases models are still too computationally expensive, especially when modeling in two or three dimensions, so in this paper we demonstrate a robust approach that drastically reduces the computational cost of evaluating a kinetic model for lignin pyrolysis. This approach is generalizable to any set of data with an underlying relationship – even if that relationship is unknown – so can be applied to experimental observations in addition to mathematical models.

Published comprehensive models for biomass pyrolysis use kinetic schemes where both the reactions and the species involved are highly lumped. (Di Blasi, 2008; White et al., 2011) Such lumped kinetic schemes have a number of limitations, but from a process development and engineering standpoint the two key limitations are the facts that lumped

models cannot predict the detailed product compositions necessary for optimizing pyrolysis reactors, nor are they accurate across ranges of reactor conditions or feedstocks.(Di Blasi, 2008; White et al., 2011) Research into developing more detailed or mechanistic kinetic schemes for biomass components to address these limitations is ongoing,(B. R. Hough et al., 2016; Vinu & Broadbelt, 2012) but the challenge of incorporating such schemes involving hundreds to thousands of reactions and species into comprehensive models, and solving them in a reasonable amount of time (on the order of hours rather than the weeks or months currently required), has not been solved.(Reyniers et al., 2015)

We tackle this challenge using our previously developed kinetic model for lignin pyrolysis.(B. R. Hough et al., 2016) Lignin – a highly cross-linked amorphous copolymer of three randomly polymerized phenylpropane units – accounts for up to 35wt% of dry biomass, and carries the highest specific energy content of the three principal components of biomass.(Azadi et al., 2013) Today, most lignin is burned for process heat,(Vishtal & Kraslawski, 2011) however the economic viability of future lignocellulosic biorefineries will depend on making lignin a high-value intermediate instead of a low-value fuel. Doing so requires the development of predictive comprehensive models that can be used to optimize fast pyrolysis processes to produce valuable renewable fuels and chemicals from lignin. As discussed earlier, existing comprehensive models lack sufficient kinetic detail to be useful for this type of optimization.

Our detailed kinetic model, involving ~100 species and 400 reactions, predicts the evolution of molecules and functional groups during lignin pyrolysis.(B. R. Hough et al., 2016) The model shows good agreement with experiment, but solving the complete set of stiff ODEs is too computationally expensive to incorporate into comprehensive particle or CFD models. Thus, the use of detailed biomass kinetics is precluded from use cases in which coupled transport and reactions must be simultaneously considered (e.g., see Paulson et al., 2014). In this work we demonstrate an approach whereby machine learning algorithms called artificial neural networks (neural nets) can learn to predict the endpoint data coming from the kinetic model with high accuracy and a computational

cost up to four orders of magnitude lower than solving the ODE model. This first step of predicting single time points at comparable accuracy and reduced speed is an important proof of concept in extending the capability of neural nets to reproduce time (or volumetric) series data in batch or plug flow type reactors, thus fulfilling a critical need in the development of a generic kinetic tool for optimization of processes with coupled transport and kinetics.

Neural nets are algorithms inspired by biological neurons for learning complex non-linear and multivariate relationships. (Bishop, 2006; Haykin, 2009) They act as a “black box” which is fed a set of input data and produces some desired output, and are widely used across disciplines for tasks like prediction, classification, and pattern recognition. Neural nets have been applied to a number of chemical engineering problems, (Asadi-Eydivand et al., 2014; Beck et al., 2016; Kayala & Baldi, 2012; Molga, 2003; Rocabruno-Valdes et al., 2015) and even to kinetic model reduction. (Conesa et al., 2004; Molga et al., 2000) In situations where detailed kinetic schemes can't be generated due to insufficient knowledge of governing mechanisms, for example when there are difficulties identifying and quantifying non-stable intermediates in a mechanism, neural nets also offer an attractive alternative. (Molga et al., 2000) Neural nets have been used in other aspects of biomass pyrolysis modeling (Mikulandrić et al., 2014; Puig-Arnavat et al., 2013; Sun et al., 2016; Sunphorka et al., 2017), but to our knowledge they have not been applied to the comprehensive modeling of biomass pyrolysis particles - where the computational cost is limiting further development.

Structurally, neural nets contain layers made up of simple processing elements called nodes that function as non-linear summing units. These nodes are interconnected with weighted connection lines between layers, where each layer receives its input from the previous layer and the output of each node feeds the next layer. The weights are adjusted iteratively during a training process, resulting in networks that can be highly accurate and extremely computationally efficient. The mathematical features of neural nets are explored in textbooks, (Bishop, 2006; Haykin, 2009) and are not described here.

2. Materials and Methods

When using neural nets only the relevant input and output signals from the kinetic model are observed while training the network. A neural net approximates the complex functional dependence of the kinetic model's system of ODEs, without any explicit knowledge of these mathematical expressions. In this work, neural networks were constructed and trained using the following steps:

1. Choice of relevant input and output variables for lignin pyrolysis
2. Generating training and validation data from the detailed kinetic model
3. Choice of neural network architecture and learning algorithms
4. Training the neural networks

2.1 Choice of input and output variables

The kinetic model that we trained neural nets to reproduce predicts the temporal evolution of molecules and functional groups during lignin pyrolysis and is described elsewhere.(B. R. Hough et al., 2016) Inputs for both the kinetic model and our neural nets are: the maximum pyrolysis temperature (T_{max}), the heating rate, and the mass fractions of carbon and hydrogen in the lignin feedstock. We trained our neural nets to predict the final values of 30 output measures after simulated pyrolysis in a batch reactor, so time was not considered as an input parameter, however we plan to extend the neural networks in future work to predict temporal changes.

The particular output measures we trained are listed in **Table S1** and describe the yields of specific functional or lumped groups; we used these outputs because they were part of a large data set already generated for previous work, but we could have used any set of outputs that the kinetic model is able to generate including concentrations or yields of individual or lumped species, or values derived from those quantities.

2.2 Generating training, validation, and test data

Training any supervised learning algorithm like a neural network involves exposing it to a sufficient amount of input-output training pairs. In order for the neural net to generalize

well it is important that this training data covers the entire range of parameter values that it should be able to predict outputs for. We used the Python package SALib to generate 250,000 parameter combinations based on the quasi-random Sobol sequence that uniformly cover the entire possible input parameter space.(Herman et al., 2015) The range of parameter values explored was 400 – 950°C for $T_{\max}$, 5 – 120,000°C/min for heating rate, 56.7 – 67.8 wt% carbon, and 5.4 – 6.6 wt% hydrogen. These ranges for carbon and hydrogen cover the allowed values in the detailed kinetic model. To generate the corresponding output data we ran the generated set of input parameters through our detailed kinetic model, resulting in 250,000 input-output learning pairs where each input contains the four parameter values, and each output contains the values of the 30 output measures.

It is common practice in supervised learning to split available learning data into three sets: a training set that is used to adjust the weights between nodes in the network; a validation set that is used to see how the trained network is performing on data that it has not been exposed to for the purpose of model selection and to decide when training is complete; and a test set to assess the performance of the final selected model on data it has never seen before. To set up these splits we first performed some pre-processing of the learning data by randomly shuffling the 250,000 learning pairs. After shuffling we divided the learning data into a training set with 140,000 input-output pairs, a validation set with 60,000 pairs, and a test set with 50,000 pairs. Then we standardized both the input and outputs to have a mean of zero and unit variance. We standardized the inputs in this way in order to prevent terms with higher magnitude, or larger ranges, from dominating the model, and to speed up convergence,(LeCun et al., 1998) and did the same to the outputs because our error function (mean squared error) is scale-sensitive so variability in output values would otherwise impact how well the net learns different outputs.

2.3 Choice of network architecture and learning algorithms

We took two approaches to designing multi-layer feed-forward neural networks that could approximate our kinetic model; one strategy was to train a single neural net to

predict all 30 output measures – which we will call the “full net” (**Figure 1A**), while the other strategy was to train 30 independent neural nets, each of which predicts a single output measure – called the “single nets” (**Figure 1B**). For all of these neural nets we included two dense hidden layers, as we found that a single hidden layer was often insufficient to adequately approximate the kinetic model (predictions for some single-layer networks resulted in unacceptably low R^2 values, around 0.5, for some output measures). To determine the optimal number of nodes in each hidden layer we carried out an exhaustive grid search over the number of nodes in each layer, within the range of 5 to 32 nodes, using the grid search tools in Scikit-learn.(Pedregosa et al., 2011) The network architectures with the lowest mean squared error on the validation set were selected for further training (**Table S2**). The number of nodes in each layer for each model is also described in **S2**.

Figure 1: Network architectures for the full net (A), and single nets (B). The input, hidden, and output variables are represented by nodes, and the weight parameters by links between nodes. Bias parameters are represented by solid nodes, x_0 and H_0^i . Subscripts n and m refer to the number of nodes in the hidden layers of the single nets (Table S2).

Activation functions in the hidden layers are hyperbolic tangents of the form recommended by LeCun et al.:(LeCun et al., 1998)

$$f(a) = 1.7159 \tanh(\frac{2}{3} a) \quad (1)$$

where a is the activation, a linear combination of the outputs from the previous layer in the network. Activation functions in the output layer are linear as is standard in regression problems. We used an adaptive learning rate algorithm, adagrad,(Duchi et al., 2011) to adjust the weights and biases to minimize the error function during the training process. Weights were initialized using Glorot weight initialization with weights sampled from a uniform distribution.(Glorot & Bengio, 2010)

2.4 Training the neural nets

After selecting the optimal network architectures from the grid search results the neural nets were trained using a batch size of 128 for a maximum of 6000 epochs. Training ended earlier if the error on the validation set increased for a sustained number of epochs. The Python package *nolearn* was used to build and train the neural nets.(Nouri, 2015)

2.5 Training a Decision Tree

In addition to the neural nets we also trained another type of machine learning model, called a decision tree. We explored the use of decision trees because they are much faster to train than neural nets (three seconds instead of one to several hours for the neural networks), and because they are less complicated so can be constructed by practitioners without the deep understanding required for training neural nets. We used the same learning data to train the decision tree as the neural networks, and followed a similar grid search procedure to optimize its architecture and determine the maximum depth and minimum number of samples in each leaf of the tree (**Table S2**). Other hyperparameters of the decision tree were assigned default values from Scikit-learn's DecisionTreeRegressor class, and Scikit-learn was used to train the tree.(Pedregosa et al., 2011) Our Python code to set up and train all of the neural nets and decision tree is available online.(B. Hough, 2016)

3. Results & Discussion

After the training procedure each machine learning model was evaluated based on how well it predicts the output of the ODE-based kinetic model. To this end each model was provided the input data from the test set, which none of the models had previously been exposed to, and model predictions were compared with the expected test set outputs.

Table 1 shows the coefficient of determination for the test set predictions of each output measure for each of the machine learning models. **Figures 2-4** are representative parity plots showing two output measures (data normalized as described above) with high R^2 values and two with low R^2 values for the single nets, full net, and decision tree, respectively. The complete set of parity plots for all output measures are available in the ESI (**Figures S1-S3**).

Table 1: Coefficients of determination (R^2) of the predicted and actual values on the test set for each of the trained machine learning models. The output measures are described in Table S1.

Output measure	Single net	Full net	Decision tree
CO	0.9990	0.9909	0.9921
CO2	0.9999	0.9938	0.9977
H-C	0.9988	0.9930	0.9959
H2O	0.9992	0.9938	0.9952
Moisture	0.9983	0.9930	0.9957
heavy_C--O	0.9965	0.9862	0.9846
heavy_aliphatic_C-C	0.9967	0.9935	0.9824
heavy_aliphatic_C-O	0.9974	0.9869	0.9879
heavy_aromatic_C-C	0.9987	0.9923	0.9892
heavy_aromatic_C-H	0.9996	0.9975	0.9970
heavy_aromatic_C-O	0.9996	0.9965	0.9972
heavy_aromatic_methoxyl	0.9995	0.9970	0.9974
heavytars	0.9981	0.9907	0.9912
light_C--O	0.9986	0.9910	0.9923
light_aliphatic_C-C	0.9997	0.9945	0.9960
light_aliphatic_C-O	0.9979	0.9860	0.9846
light_aromatic_C-H	0.9994	0.9950	0.9954
light_aromatic_C-O	0.9995	0.9954	0.9954
lighttars	0.9994	0.9979	0.9972
othergases	0.9989	0.9910	0.9957
solids	0.9982	0.9917	0.9909
tot_C--O	0.9993	0.9945	0.9979
tot_aliphatic_C-C	0.9998	0.9969	0.9979
tot_aliphatic_C-O	0.9982	0.9902	0.9933
tot_aromatic_C-C	0.9983	0.9935	0.9909
tot_aromatic_C-H	0.9995	0.9981	0.9974

tot_aromatic_C-O	0.9994	0.9965	0.9970
tot_aromatic_methoxyl	0.9994	0.9972	0.9973
totalgases	0.9992	0.9945	0.9938
totaltars	0.9984	0.9947	0.9946

Figure 2: Representative parity plots for the training, validation, and test sets of four single net models. Panels (A) and (B) present two nets with high R^2 values on the test set, while (C) and (D) are nets with relatively low R^2 values.

Figure 3: Representative parity plots for the training, validation, and test sets of four output measures in the full net. Panels (A) and (B) present two measures with high R^2 values on the test set, while (C) and (D) are measures with relatively low R^2 values.

Figure 4: Representative parity plots for the training, validation, and test sets of four output measures in the decision tree. Panels (A) and (B) present two measures with high R^2 values on the test set, while (C) and (D) are measures with relatively low R^2 values.

These figures along with **Table 1** make it clear that the single nets are able to predict all the output measures with excellent accuracy, with over 99.9% of the variation in actual values predicted by the single net models for most outputs. Both the full net and decision tree also predict the output measures very well ($R^2 > 98.2$ for all outputs), and achieve comparable performance to each other on the test set. Training the single nets results in more accurate predictions than the full net because the weights between nodes in the full net are adjusted such that they result in the best fit to all of the output measures at the

same time, whereas weights in the single nets are optimized for just a single output measure at once and thus result in better predictions for that output measure.

Histograms showing the distribution of the mean squared error (MSE) for the training, validation, and test sets are available in **Figures S4-S6**. The neural net models show a relatively even distribution of errors between the training, validation, and test sets – indicating good generalization – while the decision tree has very low error on the training set and relatively more error in the test set than the neural nets, which can be a sign of poor generalization (**Figure S6**). In our case, the very large training set compensates for this tendency of decision trees to overfit, and the spread of MSE's for the decision tree is consequently nearly the same as that of the full net. This is not the case when less training data is available (**Figure 5**), and we see that while the average MSE of the neural net on both the training and validation sets levels-off with around 35,000 training pairs, the decision tree requires 105,000 more training pairs to approach the same validation error. More advanced approaches to using decision trees, such as using them within an ensemble, have not been explored here but might mitigate this issue.

Figure 5: Learning curve depicting the change in mean squared error with the amount of available training data for the training and validation sets of a decision tree (DT) and a neural net (net) with the architectures described herein. 3-fold cross-validation was used to split the training data; lines connect the average MSE while the shaded regions represent the entire range of MSE scores at each training subset size.

The purpose of training these machine learning models was to reduce the computational cost of predicting the outputs of a complicated kinetic model so that detailed kinetics can be used in comprehensive particle modeling. To assess the reduction in computational cost we performed benchmarking tests of all the models (**Table 2**). Benchmarking accounted for the time to predict the output measures and convert them from standardized to true values. For the complete ODE model only the portion of the code where the system of ODEs is solved was included; we ignored steps to set up and write model equations. Each model was evaluated on a randomly chosen set of inputs 1000 times on a Linux workstation with two four core 2.27 GHz Intel Xeon E5607 processors and 5.7 GB of memory.

Table 2: Benchmarking results for solving the kinetic model. The time reported is the average of 1000 model calls.

Kinetic model format	Average code execution time (s)
Decision tree	1.106×10^{-4}
Full net	1.690×10^{-4}
Single net	1.747×10^{-4}
30 single nets (run serially)	4.236×10^{-3}
Complete ODE model(B. R. Hough et al., 2016)	4.725

Using the machine learning models we are able to reduce the computation expense by three to four orders of magnitude, depending on the level of accuracy that is desired and the number of output parameters required. With these speeds it is possible to evaluate detailed reaction kinetics in CFD simulations even faster than using the simple lumped ODE-based kinetics that are the current standard.

The choice of which machine learning model to use depends on the level of accuracy required and the amount of training data available. When accurate model predictions are most important, the single nets are the best choice. By itself, each single net is faster to train than the full net (though predicting the output from one trained net takes essentially the same amount of time as for the full net), and it is not necessary to predict all 30 output

measures if they are not all of interest at a given time. If there is more room for error in the predicted output measures, and a sufficient amount of training data is available such that the decision tree performs comparably to the full net, then using the decision tree is the best choice because it is faster to solve and faster to train. While not a concern on most modern computers, it should be noted that the trained decision tree is over 100 times larger on disk than a trained neural net (170MB vs. 1.3 MB).

It should be stressed that at this time the machine learning models presented in this work cannot replace full ODE model evaluation in general for the current system because our neural nets and decision tree do not account for temporal changes in the reacting system. Extending these models to include time as an additional input parameter is possible, and mathematically trivial – it would involve repeating all the steps outlined in the Experimental section, but also adding time to the four existing input parameters, so the input layer of each neural net will have five nodes instead of four. A challenge with this approach will be the large file sizes required for the learning data, because each set of our four existing inputs will need to be associated with descriptions of potentially thousands of time points, depending on the desired temporal resolution. It is also likely that neural nets for such systems will be more complex (more nodes in the hidden layers). While such nets would be more expensive to train, a one-time cost, their execution times for predicting the temporal output of the kinetic model would be similar to the benchmarking times in **Table 2**.

4. Conclusions

In this work we've outlined an approach to kinetic model reduction that reduces the computational expense of computing a detailed kinetic model for lignin pyrolysis by up to four orders of magnitude. Our neural network and decision tree models demonstrate excellent agreement with expected final values for the outputs from simulated reactions in a batch reactor. In future work we plan to extend the models to predict the temporal values of these outputs as discussed above. A logical next step in developing this framework for a biomass process optimization scheme would also be to couple the neural

net for the kinetic model within unsteady state transport equations to include the effects of heat and/or mass transport limitations.

An additional consideration is the total computational cost required to generate the model. As we demonstrated above, the neural net representation of the kinetic model is approximately four orders of magnitude faster than the average solution time using a stiff ODE solver. However, the accuracy of the model depends both on range of data represented in the training set as well as the number of training pairs used (e.g., Figure 5). Therefore, it should be straightforward to make an order of magnitude estimate of a cost vs. benefit for a given optimization of a reactor that includes spatial degrees of freedom and requires a manifold of ODE solutions across all of the mesh/grid points and expected number of optimization cycles. We also note that generation of even larger or more comprehensive training sets (e.g., including temporal resolution) could be accomplished very quickly if the ODE solver were deployed on a cloud computing or other type of large scale high performance computing resource such as AWS or a DOE leadership class computer.

The machine learning approach described is inherently generalizable and is not specific to kinetic modeling. Any set of input and output data that is based on some underlying relationship, even if that relationship is unknown, can be approximated with a neural network. Therefore, using this approach we need not fully understand the underlying physical phenomena to develop a robust, computationally efficient predictor for a given system.

Within the biomass pyrolysis realm, this means that we might skip kinetic modeling altogether, and instead generate neural networks that predict output measures of a specific experimental or industrial reactor based entirely on observed experimental data. The challenge with such an approach will be in generating sufficient training data that spans the range of conditions we might model. When sufficient data is not available ahead of time, online learning can be used to train the network as new data is generated, so as a reactor is operated the neural net predictions will improve continuously. For a

computational model, generating this data is a one-time cost after which training the neural net(s) is relatively quick (e.g., the average training time on a standard single CPU laptop is around 40 minutes for a single net), and predicting the output values from a trained network is extremely fast, as we've demonstrated. In situations where the model will only be evaluated a limited number of times, or if it does not have a pre-determined parameter range, such machine learning approaches will not improve the overall computational efficiency.

Acknowledgements

This work was supported by the National Institute for Food and Agriculture, U.S. Department of Agriculture, under Award Number 2011-68005-30416.

References

- Anca-Couce, A. (2016). Reaction mechanisms and multi-scale modelling of lignocellulosic biomass pyrolysis. *Progress in Energy and Combustion Science*, 53, 41-79.
- Anca-Couce, A., & Zobel, N. (2012). Numerical analysis of a biomass pyrolysis particle model: Solution method optimized for the coupling to reactor models. *Fuel*, 97, 80-88.
- Asadi-Eydivand, M., Solati-Hashjin, M., Farzadi, A., & Abu Osman, N. A. (2014). Artificial neural network approach to estimate the composition of chemically synthesized biphasic calcium phosphate powders. *Ceramics International*, 40, 12439-12448.
- Azadi, P., Inderwildi, O. R., Farnood, R., & King, D. A. (2013). Liquid fuels, hydrogen and chemicals from lignin: A critical review. *Renewable & Sustainable Energy Reviews*, 21, 506-523.
- Beck, D. A. C., Carothers, J. M., Subramanian, V. R., & Pfaendtner, J. (2016). Data Science: Accelerating Innovation and Discovery in Chemical Engineering. *Aiche Journal*, 62, 1402-1416.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

- Conesa, J. A., Caballero, J. A., & Reyes-Labarta, J. A. (2004). Artificial neural network for modelling thermal decompositions. *Journal of Analytical and Applied Pyrolysis*, 71, 343-352.
- Di Blasi, C. (2008). Modeling chemical and physical processes of wood and biomass pyrolysis. *Progress in Energy and Combustion Science*, 34, 47-90.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12, 2121-2159.
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics* (pp. 249-256).
- Haykin, S. S. (2009). *Neural Networks and Learning Machines*: Prentice Hall.
- Herman, J., Hadka, D., xantares, Fernando, bernardoct, Mutel, C., & Usher, W. (2015). SALib: New documentation, doc strings and installation requirements. In (0.7 ed.).
- Hough, B. (2016). lignet: artificial neural networks for modeling lignin pyrolysis. In (0.1 ed.): Zenodo.
- Hough, B. R., Schwartz, D. T., & Pfaendtner, J. (2016). Detailed kinetic modeling of lignin pyrolysis for process optimization. *Industrial & Engineering Chemistry Research*.
- Kayala, M. A., & Baldi, P. (2012). ReactionPredictor: Prediction of Complex Chemical Reactions at the Mechanistic Level Using Machine Learning. *Journal of Chemical Information and Modeling*, 52, 2526-2540.
- LeCun, Y., Bottou, L., Orr, G. B., & Muller, K. R. (1998). Efficient backprop. *Neural Networks: Tricks of the Trade*, 1524, 9-50.
- Mehrabian, R., Zahirovic, S., Scharler, R., Obernberger, I., Kleditzsch, S., Wirtz, S., Scherer, V., Lu, H., & Baxter, L. (2012). A CFD model for thermal conversion of thermally thick biomass particles. *Fuel Processing Technology*, 95, 96-108.
- Mikulandrić, R., Lončar, D., Böhning, D., Böhme, R., & Beckmann, M. (2014). Artificial neural network modelling approach for a biomass gasification process in fixed bed gasifiers. *Energy Conversion and Management*, 87, 1210-1223.
- Moghtaderi, B. (2006). The state-of-the-art in pyrolysis modelling of lignocellulosic solid fuels. *Fire and Materials*, 30, 1-34.
- Molga, E. J. (2003). Neural network approach to support modelling of chemical reactors: problems, resolutions, criteria of application. *Chemical Engineering and Processing*, 42, 675-695.
- Molga, E. J., van Woezik, B. A. A., & Westerterp, K. R. (2000). Neural networks for modelling of chemical reaction systems with complex kinetics: oxidation of 2-octanol with nitric acid. *Chemical Engineering and Processing*, 39, 323-334.
- Nouri, D. (2015). nolearn. In (0.7 ed., pp. Abstractions around neural net libraries, most notably Lasagne.). Github.
- Papadikis, K., Gu, S., Bridgwater, A. V., & Gerhauser, H. (2009). Application of CFD to model fast pyrolysis of biomass. *Fuel Processing Technology*, 90, 504-512.
- Paulsen, A. D., Hough, B. R., Williams, C. L., Teixeira, A. R., Schwartz, D. T., Pfaendtner, J., & Dauenhauer, P. J. (2014). Fast Pyrolysis of Wood for Biofuels:

- Spatiotemporally Resolved Diffuse Reflectance In situ Spectroscopy of Particles. *Chemosuschem*, 7, 765-776.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Puig-Arnabat, M., Hernández, J. A., Bruno, J. C., & Coronas, A. (2013). Artificial neural network models for biomass gasification in fluidized bed gasifiers. *Biomass and Bioenergy* 49, 279-289.
- Ragauskas, A. J., Williams, C. K., Davison, B. H., Britovsek, G., Cairney, J., Eckert, C. A., Frederick, W. J., Jr., Hallett, J. P., Leak, D. J., Liotta, C. L., Mielenz, J. R., Murphy, R., Templer, R., & Tschaplinski, T. (2006). The path forward for biofuels and biomaterials. *Science*, 311, 484-489.
- Reyniers, P. A., Schietekat, C. M., Van Cauwenberge, D. J., Vandewalle, L. A., Van Geem, K. M., & Marin, G. B. (2015). Necessity and Feasibility of 3D Simulations of Steam Cracking Reactors. *Industrial & Engineering Chemistry Research*, 54, 12270-12282.
- Rocabruno-Valdes, C. I., Ramirez-Verduzco, L. F., & Hernandez, J. A. (2015). Artificial neural network models to predict density, dynamic viscosity, and cetane number of biodiesel. *Fuel*, 147, 9-17.
- Sun, Y., Liu, L., Wang, Q., Yang, X., & Tu, X. (2016). Pyrolysis products from industrial waste biomass based on a neural network model. *Journal of Analytical and Applied Pyrolysis* 120, 94-102.
- Sunphorka, S., Chalermssinsuwan, B., & Piumsomboon, P. (2017). Application of artificial neural network for kinetic parameters prediction of biomass oxidation from biomass properties. *Journal of the Energy Institute*, 90, 51-61.
- Vinu, R., & Broadbelt, L. J. (2012). A mechanistic model of fast pyrolysis of glucose-based carbohydrates to predict bio-oil composition. *Energy & Environmental Science*, 5, 9808-9826.
- Vishtal, A., & Kraslawski, A. (2011). Challenges in industrial applications of technical lignins. *Bioresources*, 6, 3547-3568.
- White, J. E., Catallo, W. J., & Legendre, B. L. (2011). Biomass pyrolysis kinetics: A comparative critical review with relevant agricultural residue case studies. *Journal of Analytical and Applied Pyrolysis*, 91, 1-33.